TOWARD AN H-INDEPENDENT ALGEBRAIC MULTIGRID METHOD FOR MAXWELL'S EQUATIONS

JONATHAN J. HU[3,5], RAYMOND S. TUMINARO[3,5], PAVEL B. BOCHEV[1,5], CHRISTOPHER J. GARASI[4,5], AND ALLEN C. ROBINSON[2,5]

Abstract. We propose a new algebraic multigrid (AMG) method for solving the eddy current approximations to Maxwell's equations. This AMG method has its roots in an algorithm proposed by Reitzinger and Schöberl. The main focus in the Reitzinger and Schöberl method is to maintain null-space properties of the weak $\nabla \times \nabla \times$ operator on coarse grids. While these null-space properties are critical, they are not enough to guarantee h-independent convergence rates of the overall multigrid scheme. We present a new strategy for choosing intergrid transfers that not only maintains the important null-space properties on coarse grids but also yields significantly improved multigrid convergence rates. This improvement is related to those we explored in a previous paper, but is fundamentally simpler, easier to compute, and performs better with respect to both multigrid operator complexity and convergence rates. The new strategy builds on ideas in smoothed aggregation to improve the approximation property of an existing interpolation operator. By carefully choosing the smoothing operators, we show how it is sometimes possible to achieve h-independent convergence rates with a modest increase in multigrid operator complexity. Though this ideal case is not always possible, the overall algorithm performs significantly better than the original scheme in both iterations and run time. Finally, the Reitzinger and Schöberl method, as well as our previous smoothed method, are shown to be special cases of this new algorithm.

Key words. Maxwell's equations, eddy currents, AMG, multigrid, smoothed aggregation.

AMS subject classifications. 76D05, 76D07, 65F10, 65F30

1. Introduction.

We consider the 3D eddy current formulation of Maxwell's equations:

$$\nabla \times \nabla \times \vec{E} + \sigma \vec{E} = \vec{f}, \tag{1.1}$$

with Neumann, Dirichlet, or periodic boundary conditions, where $\vec{E}$ is the unknown electric field to be computed, σ is the spatially-varying electrical conductivity, and $\vec{f}$ is the known right-hand side. We assume that (1.1) is discretized with first-order edge elements on arbitrary unstructured meshes so that the following vector identity is preserved in a discrete sense:

$$\nabla \times (\nabla \phi) = 0. \tag{1.2}$$

This identity states that gradients of scalar functions lie within the kernel of the curl operator. The discrete version of (1.1) is denoted by

$$K_1^{(e)} e_1 = f_1, \tag{1.3}$$

[1]Sandia National Laboratories, Computational Math/Algorithms, P.O. Box 5800, MS 1110, Albuquerque, NM 87185 (pbboche@sandia.gov).

[2]Sandia National Laboratories, Computational Physics R&D, P.O. Box 5800, MS 0378, Albuquerque, NM 87185 (acrobin@sandia.gov).

[3]Sandia National Laboratories, Computational Math/Algorithms, PO Box 969, MS 9159, Livermore, CA 94551 (jhu@sandia.gov, rstumin@sandia.gov)

[4]Sandia National Laboratories, HEDP Theory & ICF Target Design, P.O. Box 5800, MS 1186, Albuquerque, NM 87185 (cjgaras@sandia.gov).

[5]Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000.

where $K_1^{(e)} = S_1 + M_1$. S_1 is the discrete approximation to the weak form of the first term in (1.1), and M_1 is the discrete approximation to the weak form of the second term in (1.1). The discrete null-space analog of (1.2) is represented by the rectangular matrix T_1. This matrix is a discrete gradient operator and is trivial to construct. In particular, each row of T_1 contains at most two nonzeros (± 1) and corresponds to an edge between two nodes in the associated nodal graph [5]. The following relationship holds:

$$S_1 T_1 = \Theta,$$

where Θ denotes the zero matrix. In this paper, we use the notation (*curl,curl*) to refer to S_1 as it emphasizes the character of the operator. Edge element discretizations of the eddy current formulation with the discrete null-space property have been investigated for both tetrahedral and hexahedral meshes. While the multigrid techniques described here are not restricted to hexahedral meshes, all of our experiments have been with hexahedral meshes, where it is assumed that the domain of interest can in fact be covered by such elements [4, 16].

The focus of this paper is the solution of the discrete linear system given in (1.3) and the special treatment required by the multigrid smoother and the multigrid grid transfers to address the large (*curl,curl*) kernel. Several groups have considered this problem. Hiptmair developed a geometric multigrid method with h-independent (or mesh-independent) convergence rates in [10]. Reitzinger and Schöberl proposed an initial algebraic multigrid (AMG) method for Maxwell's equations [11] that maintains a coarse grid notion of (1.2) but does not have mesh-independent convergence rates. We considered modifications to Reitzinger and Schöberl's strategy for choosing a prolongator in a previous paper [3]. The current paper proposes an improvement to the intergrid transfers that is related to those explored in [3]. The resulting method performs noticeably better than the previous method in terms of both multigrid cost per iteration and convergence. In addition, the new prolongator is much simpler to understand and implement and leads to nearly h-independent convergence of a multigrid W-cycle on some model problems.

2. Background. In this section, we give an overview of multigrid, as well as previous work in applying multigrid to Maxwell's equations.

2.1. Multigrid Overview. Multigrid methods (e.g., [9], [12], [8]) are among the most efficient iterative algorithms for solving the linear system, $Ax = b$, associated with elliptic partial differential equations. The basic idea is to damp errors by utilizing multiple resolutions in the iterative scheme. High-energy (or oscillatory) components are effectively reduced through a simple smoothing procedure, while the low-energy (or smooth) components are tackled using an auxiliary lower resolution version of the problem (coarse grid). The idea is applied recursively on the next coarser level. An example multilevel iteration is given in Figure 2.1 to solve

$$A_1 u_1 = b_1. \tag{2.1}$$

The two operators needed to specify the multigrid method fully are the relaxation procedures, R_k, $k = 1, \ldots, N_{levels}$, and the grid transfers, P_k, $k = 2, \ldots, N_{levels}$. Note that P_k is an interpolation operator that transfers grid information from level k to level $k-1$. The coarse grid discretization operator A_{k+1} ($k \geq 1$) is specified by the Galerkin product

$$A_{k+1} = P_{k+1}^T A_k P_{k+1}. \tag{2.2}$$

```
// Solve A_k u_k = b_k
procedure multilevel(A_k, b_k, u_k, k)
    u_k = R_k(A_k, b_k, u_k);
    if ( k ≠ N_levels)
        r_k = b_k − A_k u_k ;
        A_{k+1} = P_{k+1}^T A_k P_{k+1};
        u_{k+1} = 0;
        multilevel(A_{k+1}, P_{k+1}^T r_k, u_{k+1}, k + 1);
        u_k = u_k + P_{k+1} u_{k+1};
        u_k = R_k(A_k, b_k, u_k);
```

FIG. 2.1. *Multigrid V-cycle consisting of N_{levels} grids to solve $A_1 u_1 = b_1$.*

The key to fast convergence is the complementary nature of these two operators. That is, errors not reduced by R_k must be well interpolated by P_k.

When grid transfers are defined using mesh and/or finite element information, this is generally termed geometric multigrid. While geometric multigrid methods often work well, they can be somewhat difficult to implement for highly unstructured meshes with irregular boundaries (especially on parallel machines). Further, they often rely on details of the specific simulation program. That is, applications are forced to develop their own version of geometric multigrid where they supply a mesh hierarchy and a means to move between grids. While potentially efficient, software development can be time consuming and the resulting multigrid code may not be well suited for other applications. In algebraic multigrid, however, only A_1 and b_1 are given, hence R_k and P_k must be deduced from purely algebraic principles. While constructing multigrid methods via algebraic concepts presents certain challenges, AMG can be used for several problem classes without requiring a major effort for each application. In this paper, we focus on a strategy to determine the P_k's based on algebraic principles. It is assumed that A_1 and b_1 are given.

2.2. Geometric Multigrid for Maxwell's Equations. A unique difficulty in solving Maxwell's equations arises from the kernel of the (*curl,curl*) operator. This kernel includes the gradients of all differentiable scalar functions. Since the gradient of a smooth function is smooth and the gradient of an oscillatory function is oscillatory, it is clear that both smooth and oscillatory functions lie within the kernel. We denote the range space of operator A by $\mathcal{R}an(A)$. In the standard multigrid context, relaxation must damp oscillatory errors in $\mathcal{R}an(T_1)$ and $\mathcal{R}an(T_1^{\perp})$, and the coarse grid correction must approximate smooth errors in $\mathcal{R}an(T_1)$ and $\mathcal{R}an(T_1^{\perp})$. While the canonical coarse grid transfers[1] of a geometric multigrid method sufficiently approximate smooth errors in $\mathcal{R}an(T_1)$ and $\mathcal{R}an(T_1^{\perp})$, a special relaxation procedure must be used so that oscillatory errors in $\mathcal{R}an(T_1)$ are damped. This is because methods like Jacobi and Gauss-Seidel rely on oscillatory error components being well represented in the residual. When M_1 is small compared to S_1, however, errors in the kernel of the (*curl,curl*) operator are not well represented. This can be seen by considering the

[1] Using the edge element basis functions.

procedure distributed_smoother $(K_1^{(e)}, b, u)$
Smooth on $K_1^{(e)} u = b$
$r = b_1 - K_1^{(e)} u$
$\tilde{K}_1 = T_1^T K_1^{(e)} T_1 \qquad // = T_1^T M_1 T_1$
$\tilde{b} = T_1^T r, \;\; \tilde{u} = 0$
Smooth on $\tilde{K}_1 \tilde{u} = \tilde{b}$
$u = u + T_1 \tilde{u}$

FIG. 2.2. *Distributed Smoother for system (1.3).*

following error decomposition:

$$\begin{array}{rcll} \epsilon & = & \epsilon_T + \epsilon_{T^\perp} & \epsilon_T \in \mathcal{R}an(T_1),\ \epsilon_{T^\perp} \in \mathcal{R}an(T_1^\perp),\ ||\epsilon_T|| \approx ||\epsilon_{T^\perp}|| \\ r & = & (S_1 + M_1)(\epsilon_T + \epsilon_{T^\perp}) & \\ & = & S_1 \epsilon_{T^\perp} + M_1(\epsilon_T + \epsilon_{T^\perp}) & \\ & \approx & S_1 \epsilon_{T^\perp} \text{ when } ||S_1|| \gg ||M_1||. & \end{array} \tag{2.3}$$

Hiptmair was essentially the first to propose a suitable relaxation procedure for (1.3), and this led to a geometric multigrid method with h-independent convergence rates [10]. This smoother is a specific instance of a more general idea termed distributed relaxation [7]. Specifically, a standard smoother is first applied to $K_1^{(e)}$. Then, a correction equation is formed and projected onto $\mathcal{R}an(T_1)$. Standard smoothing is applied to the projected equation, and the correction is added into the previous solution estimate. A summary of this smoother is given in Figure 2.2. It is important to note that a basis for the kernel of the (*curl,curl*) operator must be explicitly constructed on all levels within a multigrid scheme to apply this relaxation procedure. A related smoother was considered in [15] for mixed finite element problems, and alternative smoothers have been proposed by [2].

2.3. AMG for Maxwell's Equations. While several good geometric multigrid options now exist, the situation is not nearly so clear for algebraic multigrid methods. The first practical AMG method for solving Maxwell's equations was developed by Reitzinger and Schöberl [11]. The principal idea of the method is to maintain the representation of a discrete gradient and its relationship to the discrete (*curl,curl*) operator on coarse meshes. Reitzinger and Schöberl achieve this by creating two multigrid hierarchies. The primary hierarchy is a coarsening of the space where the problem (1.3) lies. The interpolation operator is given by $P_k^{(e)}$, and the coarse grid operator is given by the Galerkin product

$$K_{k+1}^{(e)} = (P_k^{(e)})^T K_k^{(e)} P_k^{(e)}.$$

This is the hierarchy that is used during the multigrid iterations. The other hierarchy serves an auxiliary role and is used only in the setup phase to construct the intergrid transfers $P_k^{(e)}$ for the primary hierarchy.

To construct the auxiliary hierarchy, Reitzinger and Schöberl discretize the partial differential equation

$$\int_\Omega \sigma u \cdot v + \int_\Omega \nabla u \cdot \nabla v \tag{2.4}$$

using nodal finite elements on the same mesh as (1.1). This yields a matrix $K_1^{(n)}$. The matrix T_1 is also needed for this algebraic multigrid method. T_1 can be created directly from the mesh. This is done by adding one matrix row for each edge in $K_1^{(e)}$ with the two nonzero entries corresponding to nodal degrees of freedom associated with the mesh. It is also possible, however, to construct T_1 using the matrix graph of $K_1^{(n)}$ [5]. Specifically, each off-diagonal nonzero (i, j) in the upper triangular portion of $K_1^{(n)}$ corresponds to a row in T_1 containing a 1 and a -1 in columns i and j. This implies that one way to build a coarse grid discrete "gradient", T_2, is to apply an algebraic multigrid procedure to $K_1^{(n)}$. This yields a prolongation operator, $P_k^{(n)}$, and a coarse grid discretization matrix, $K_2^{(n)} = (P_1^{(n)})^T K_1^{(n)} P_1^{(n)}$. The graph of $K_2^{(n)}$ can then be used to define a coarse grid discrete gradient.

The auxiliary nodal hierarchy can be viewed as an explicit coarsening of the (*curl,curl*) null space. The key question is now how to choose the interpolation, $P_1^{(e)}$, so that T_2 is precisely the null space of the coarse grid discrete (*curl,curl*) operator,

$$S_2 = (P_1^{(e)})^T S_1 P_1^{(e)}. \tag{2.5}$$

Reitzinger and Schöberl showed that the following commuting relationship is necessary and sufficient to maintain the null-space property on coarse grids:

$$P_k^{(e)} T_{k+1} = T_k P_k^{(n)}, \tag{2.6}$$

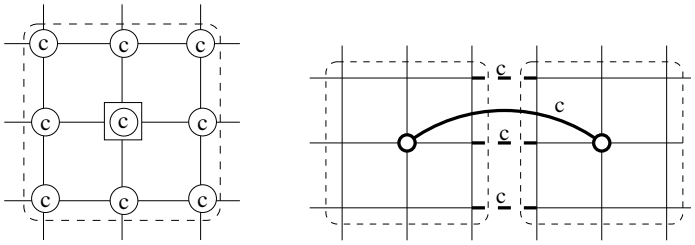
where $T_k, P_k^{(n)}$, and $P_k^{(e)}$ are the discrete gradient, nodal prolongator, and edge prolongator on the k^{th} level. Specifically, assume that $S_k T_k = \Theta$ and that (2.6) holds. Then,

$$\begin{aligned} S_{k+1} T_{k+1} &= (P_k^{(e)})^T S_k P_k^{(e)} T_{k+1} \\ &= (P_k^{(e)})^T S_k T_k P_k^{(n)} \qquad \text{by (2.6)} \\ &= \Theta. \end{aligned}$$

Further discussions of the commuting property (2.6) are given in [11]. Thus, our computational task is reduced to producing a $P_k^{(e)}$ given T_k, T_{k+1}, and $P_k^{(n)}$ such that (2.6) holds. This will preserve the null space on coarse meshes and allow the distributed relaxation procedure to be applied. To do this effectively, however, a prolongation operator that maintains good approximation properties and leads to low multigrid operator complexities (cost per multigrid iteration) is necessary.

Reitzinger and Schöberl proposed a strategy for computing a $P_k^{(e)}$ that satisfies (2.6) and gives low multigrid operator complexities. Specifically, coarse nodes are first constructed by aggregating (or grouping) fine nodes together. $P_k^{(n)}$ is now defined as piecewise-constant interpolation from each coarse node to all of the fine nodes within its aggregate. A coarse nodal discretization is then defined using $P_k^{(n)}$, $K_k^{(n)}$ and the Galerkin product, (2.2). Coarse edges in the coarse nodal matrix simply link coarse nodes from adjacent aggregates. $P_k^{(e)}$ is then defined by injecting the value from each coarse edge to all the fine edges connecting the two adjacent aggregates associated with the coarse edge. We omit the details of the prolongator and refer the reader to [11] and [3]. Figure 2.3 depicts a simple example for a single nodal basis function and a single edge basis function in the respective prolongators.

Reitzinger and Schöberl's paper properly identified the importance of (2.6). Their AMG method is easy to compute, maintains low multigrid operator complexity, and



(a) Node interpolation.

(b) Edge interpolation.

FIG. 2.3. *(a) Reitzinger and Schöberl piecewise-constant nodal interpolation. The coarse grid value (in square) is interpolated only to nodes that form the aggregate (in circles). (b) Interpolation on a uniform mesh within a single edge of $P_k^{(e)}$. The coarse grid edge (heavy solid) with value c is interpolated only to fine edges (heavy dashed) passing between nodal aggregates (in dashed boxes).*

Grid Size	CG/AMG iterations
15×15	16
25×25	24
50×50	42
100×100	76
150×150	93

TABLE 2.1

Number of iterations to reduce initial residual by 10^{10} with the Reitzinger and Schöberl AMG method applied to a model problem on the unit square with $\sigma = 10^{-3}$.

is relatively insensitive to the electric conductivity. The main drawback, however, is that the prolongator has poor approximation properties, and so multigrid convergence rates suffer. In particular, h-independent algebraic multigrid convergence rates typically rely on accurately approximating functions with smaller energy norm (i.e. algebraically smooth functions) on coarse meshes. Within smoothed aggregation, for example, the theory relies on bounding the energy of all the coarser grid basis functions independent of mesh spacing. This is not possible with piecewise-constant basis functions. The Reitzinger and Schöberl edge prolongation operators are built using piecewise-constant interpolation for the nodal space and actually are less accurate than piecewise constant. In fact, when interpolating coarse grid edges, there are no contributions to fine grid edges that are contained completely within an aggregate. Contributions are made only to fine grid edges that connect the two aggregates defining the coarse grid edge. Thus, it is not possible to bound the energy of the coarse grid basis functions and so it is not surprising that the resulting method has convergence rates that are far from h-independent. Table 2.1 illustrates the growth in iterations as the mesh size increases for a model two-dimensional problem using a preconditioned conjugate gradient algorithm where a Reitzinger and Schöberl AMG method is used in a V(1,1) cycle.[2] The smoother consists of three steps: one symmetric Gauss-Seidel iteration on the entire system, one symmetric Gauss-Seidel iteration on the null-space

[2] V(1,1) indicates a V-cycle multigrid algorithm is used with 1 pre-smoothing sweep and 1 post-smoothing sweep on each level.

projected equations, and a final symmetric Gauss-Seidel iteration on the entire system. It is clear for very large problems that this lack of scalability will be prohibitive. Therefore, the remainder of this paper describes a new prolongation operator that improves scalability.

3. An Improved Prolongation Operator. Consider the new prolongation operator

$$\mathcal{P}_k^{(e)} = (I - \alpha D_{S,k}^{-1} S_k + \beta T_k D_{T,k}^{-1} T_k^T M_k) P_k^{(e)}, \tag{3.1}$$

where $P_k^{(e)}$ is the Reitzinger and Schöberl edge prolongator, S_k is the (*curl,curl*) term on the kth level, M_k is the mass term on the kth level, $D_{S,k}$ is the diagonal of S_k, $D_{T,k}$ is the diagonal of $T_k^T M_k T_k$, and α and β are damping parameters. It is easy to show that $\mathcal{P}_k^{(e)}$ still satisfies the commuting properties. Namely,

$$\begin{aligned}
\mathcal{P}_k^{(e)} T_{k+1} &= (I - \alpha D_{S,k}^{-1} S_k + \beta T_k D_{T,k}^{-1} T_k^T M_k) P_k^{(e)} T_{k+1} \\
&= (I - \alpha D_{S,k}^{-1} S_k + \beta T_k D_{T,k}^{-1} T_k^T M_k) T_k P_k^{(n)} \qquad \text{by (2.6)} \\
&= (I + \beta T_k D_{T,k}^{-1} T_k^T M_k) T_k P_k^{(n)} \qquad \text{using } S_k T_k = \Theta \\
&= T_k (I + \beta D_{T,k}^{-1} T_k^T M_k T_k) P_k^{(n)} \\
&= T_k \mathcal{P}_k^{(n)},
\end{aligned}$$

where

$$\mathcal{P}_k^{(n)} = (I + \beta D_{T,k}^{-1} T_k^T M_k T_k) P_k^{(n)}. \tag{3.2}$$

Thus, the commuting property is satisfied when the nodal prolongator is given by $\mathcal{P}_k^{(n)}$.

The new nodal prolongator (3.2) can be viewed as a *smoothed* version of the original piecewise-constant nodal interpolation operator. This terminology comes from the smoothed aggregation multigrid method [14, 13]. In smoothed aggregation multigrid, an original (or tentative) prolongator is developed. Then, a new prolongator, Q, is produced by lowering the energy of the tentative prolongator via the following damped Jacobi iteration:

$$Q = (I - \gamma D^{-1} A) Q_T, \tag{3.3}$$

where Q_T is a tentative prolongator, A is some fine grid discretization matrix, D is the diagonal of A, and γ is a damping parameter. When applied to a Poisson operator, the tentative prolongator corresponds to piecewise-constant interpolation. For this paper, it is important to understand that the smoothed prolongator has significantly better approximation properties than piecewise-constant interpolation and that the smoothing step is critical to obtaining h-independent multigrid convergence rates for the nodal problem [6, 13, 14].

In our context, it is reasonable to expect that a smoothed nodal prolongator *might* give rise to an improved edge prolongator because low-energy modes are better approximated. If the nodal problem is given by

$$K_1^{(n)} = T_1^T K_1^{(e)} T_1 = T_1^T M_1 T_1 \tag{3.4}$$

instead of discretizing (2.4), then Equations (3.2) and (3.3) are now identical with $\gamma = \beta$ and $A = K_1^{(n)}$. That is, the new nodal prolongator corresponds to a *smoothed* version of the original piecewise-constant prolongator. Using (3.4) has the distinct advantage that the user no longer needs to provide a separate nodal discretization matrix. Within the multigrid method, matrix $K_1^{(n)}$ can be constructed automatically as described in (3.4) by operators that are available. In fact, $K_1^{(n)}$ also appears in the distributed relaxation method in Figure 2.2 and so is already computed and available within the method. Additionally, the operator $T_k^T M_k T_k$ still incorporates material properties, σ, which are advantageous to utilize in the nodal aggregation phase.

It is now clear that the $\beta T_k D_{T,k}^{-1} T_k^T M_k$ term in (3.1) corresponds to smoothing the nodal prolongator. To complete our understanding of this term, however, it is necessary to look at the important role the nodal prolongator plays in the AMG iterations (even though it does not appear explicitly). To this end, we consider the Galerkin product and the coarse grid nodal matrix obtained by projecting the coarse edge operator into the null space. It is important to keep in mind that this matrix (which is used in relaxation) governs convergence in the null space. Specifically, we have

$$\begin{aligned} K_{k+1}^{(n)} &= (\mathcal{P}_k^{(n)})^T K_k^{(n)} \mathcal{P}_k^{(n)} \\ &= (\mathcal{P}_k^{(n)})^T T_k^T K_k^{(e)} T_k \mathcal{P}_k^{(n)} \qquad \text{by (3.4)} \\ &= T_{k+1}^T (\mathcal{P}_k^{(e)})^T K_k^{(e)} \mathcal{P}_k^{(e)} T_{k+1} \qquad \text{by (2.6)} \\ &= T_{k+1}^T K_{k+1}^{(e)} T_{k+1} \end{aligned}$$

This implies that it is, in fact, the nodal interpolation operator that completely determines the null-space properties of the discretization matrix on the coarse meshes. Thus, an improved nodal prolongator improves the coarse discretization within the null space.

Before concluding this discussion of nodal prolongators, it is important to make one modification to (3.1). In particular, the M_k term can be simplified to significantly reduce the nonzeros in both $\mathcal{P}^{(e)}$ and the resulting coarse grid discretization matrices. To do this, we replace M_k by $\tilde{M}_k$, where $\tilde{M}_k$ is obtained by lumping the off-diagonal elements in M_k to the diagonal. This is a well-known technique in approximating mass matrices. In our context, it can be helpful in reducing the number of nonzeros and does not affect the commuting property. As a result, we now have

$$\mathcal{P}_k^{(e)} = (I - \alpha D_{S,k}^{-1} S_k + \beta T_k D_{T,k}^{-1} T_k^T \tilde{M}_k) P_k^{(e)}. \tag{3.5}$$

The $\alpha D_{S,k}^{-1} S_k$ term in (3.5) does not appear in the new nodal prolongator. In fact, for a given nodal prolongator there are several possible edge prolongators that satisfy the commuting property. To better understand the $\alpha D_{S,k}^{-1} S_k$ term, we rewrite (3.5) as

$$\begin{aligned} \mathcal{P}_k^{(e)} &= (I - \alpha D_{S,k}^{-1} S_k)(I + \beta T_k D_{T,k}^{-1} T_k^T \tilde{M}_k) P_k^{(e)} \\ &= (I - \alpha D_{S,k}^{-1} S_k) \hat{\mathcal{P}}_k^{(e)}. \end{aligned} \tag{3.6}$$

It can now be seen that $\hat{\mathcal{P}}_k^{(e)}$ is essentially a tentative edge prolongator obtained using the smoothed nodal prolongator. To improve $\hat{\mathcal{P}}_k^{(e)}$, a further smoothing step in the edge space is performed to arrive at $\mathcal{P}_k^{(e)}$. This additional smoothing step is

needed because the edge prolongator may not be as accurate (or as smooth) as the corresponding nodal prolongator.

To better understand the relationship between nodal and smoothed prolongators, consider an operator, $Q^{(n)}$, that corresponds to linear nodal interpolation on a uniform mesh. Further, let us assume that we have obtained the operator $Q^{(e)}$ satisfying the commuting equation:

$$Q^{(e)}T_{k+1} = T_k Q^{(n)}. \tag{3.7}$$

If the right-hand side of (3.7) is applied to a coarse grid linear function v, then the resulting function is essentially constant. This follows from the fact that $Q^{(n)}v$ produces a linear function; a discrete gradient T_k applied to $Q^{(n)}v$ yields a constant function. Combining this with (3.7) implies that

$$Q^{(e)}z = w, \tag{3.8}$$

where w is a constant and $z = T_{k+1}v$ (implying that z is a constant function). Clearly, there are several $Q^{(e)}$ that satisfy (3.8), including inaccurate piecewise-constant edge interpolation! The important conclusion to draw from this fact is that the commuting relationship does not imply that the edge prolongator is as accurate as the corresponding nodal prolongator. In our context, this has been observed experimentally and motivates the application of an additional edge smoothing step (3.6) to improve the properties of the edge prolongator.

To complete the prolongator description (3.5), we must choose the damping parameters α and β. These parameters are chosen according to standard smoothed aggregation multigrid criteria, based on energy minimization [13]. Specifically,

$$\alpha = \frac{4}{3}\rho(D_{S,k}^{-1}S_k)^{-1}, \tag{3.9}$$

and

$$\beta = \frac{4}{3}\rho(D_{T,k}^{-1}T_k^T M_k T_k)^{-1}, \tag{3.10}$$

where $\rho(\cdot)$ is the spectral radius. This spectral radius is estimated using a few (e.g. 5-10) iterations of a symmetric eigenvalue solver (as in standard smoothed aggregation).

4. AMG Operator Complexity. The main AMG difficulty in solving (1.1) is that the grid transfers must satisfy the commuting relationship and, at the same time, be sufficiently accurate in both the null space and its complement. The remarkable aspect of (3.5) is that it improves the accuracy of the Reitzinger and Schöberl prolongator in both the null space and its complement *without* adversely affecting the overall cost per iteration of the AMG process. To see this, we first define multigrid operator complexity as follows:

$$\text{AMG operator complexity} = \frac{\sum_{k=1}^{N_{levels}} nnz(K_k^{(e)})}{nnz(K_1^{(e)})}. \tag{4.1}$$

where $nnz(A)$ is the number of nonzeros in the matrix A. Ideally, $nnz(K_k^{(e)}) \ll nnz(K_1^{(e)})$ for $k > 1$. This occurs when coarse grid matrices have significantly fewer rows and only a modest increase in the number of nonzeros per row relative to $K_1^{(e)}$.

To analyze nonzero behavior, we look more closely at the coarse grid discretization matrix given by the Galerkin formula:

$$K_{k+1}^{(e)} = (\mathcal{P}_k^{(e)})^T K_k^{(e)} \mathcal{P}_k^{(e)}.$$

Substituting a simplified version of (3.5), we define

$$\hat{K}_{k+1}^{(e)} = (P_k^{(e)})^T (I - S_k + Z_k) K_k^{(e)} (I - S_k + Z_k) P_k^{(e)},$$

where $Z_k = T_k T_k^T$. $\hat{K}_{k+1}^{(e)}$ is identical to $K_{k+1}^{(e)}$ when the damping parameters are one, and the diagonal matrices are the identity in (3.5). These simplifications obviously do not affect multigrid operator complexity. If we multiply the three inner terms and keep only the higher order terms, we have

$$\tilde{K}_{k+1}^{(e)} = (P_k^{(e)})^T (S_k K_k^{(e)} S_k - S_k K_k^{(e)} Z_k - Z_k K_k^{(e)} S_k + Z_k K_k^{(e)} Z_k) P_k^{(e)}. \tag{4.2}$$

Once again, the nonzero pattern of $\tilde{K}_{k+1}^{(e)}$ is identical to that of $K_{k+1}^{(e)}$ as the higher order terms govern this pattern. The multigrid operator complexity of the AMG iteration is connected to understanding how the number of nonzeros per row grows in the above expression. For general unstructured matrices this is quite cumbersome. However, for simple restricted cases it is possible to show how the number of nonzeros behave. Specifically, consider a regular uniform mesh on a square. Let us further assume that each aggregate is a perfect 3×3 brick (as depicted in Figure 2.3). Typically, aggregates are defined by some greedy algorithm. A center node is picked, preferably one not adjacent to existing aggregates, and then this node along with its immediate neighbors defines a new aggregate. In the ideal case, perfect 3×3 bricks will be formed when the discretization of (2.4) on a uniform grid yields a 9-point stencil or when $K^{(n)}$ has the form (3.4) on a uniform grid. Though this case is idealized, it gives us insight into the multigrid operator complexity growth we might expect in more realistic settings. Clearly, if the number of nonzeros grows drastically in the ideal case, one should expect high multigrid operator complexity in the non-ideal situation.

For the ideal case, it is possible to illustrate the support of a single basis function (or column) in $P_k^{(e)}$ and show how this support grows with the four terms in (4.2). Specifically, consider a vector, v, with a set of nonzero edge entries denoted E_v. These edges are incident to a set of nodes, denoted N_v, and a set of grid cells, denoted C_v.

The application of S_k or $K_k^{(e)}$ to v results in a new vector with nonzero elements in any edge contained in the cells C_v. The application of Z_k to v results in nonzero elements in any edge incident to N_v. Figures 4.1-4.4 show the multigrid operator complexity details. The three vertical lines in the center represent the j^{th} column of $P_k^{(e)}$ and the additional lines illustrate the support growth. The key point to notice in the figures is that the support never touches bordering aggregates (aggregates that are only partially illustrated). This implies that the j^{th} column of $\tilde{K}_{k+1}^{(e)}$ is zero for all coarse grid edges incident to these bordering aggregates. Thus, there are only seven nonzeros in the j^{th} column corresponding to the seven coarse edges that connect the six central aggregates depicted in the figures. All of this means that the nonzeros per row do not grow in this simple case as the original fine grid stencil contains seven nonzeros. In three dimensions, the situation is identical when each aggregate corresponds to a $3 \times 3 \times 3$ brick. Here, there is once again no growth in the number of nonzeros per row. Of course, ideal aggregates will not generally occur. However, we have some confidence that the multigrid operator complexity will not

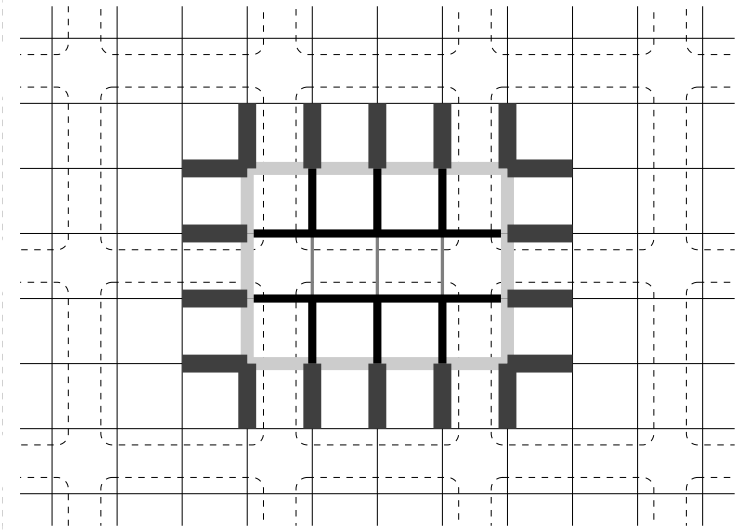


FIG. 4.1. *Nonzero edges due to* $Z_k K_k^{(e)} Z_k P_k^{(e)}$ *term.* —— *denotes the three nonzero edges in* $P_k^{(e)} e_j$ *where* e_j *has only one nonzero in the* j^{th} *position.* ▬ *denotes additional nonzero edges in* $Z_k P_k^{(e)} e_j$. ▬ *denotes the extra nonzero edges in* $K_k^{(e)} Z_k P_k^{(e)} e_j$ *and finally* ▬ *denotes the remaining nonzero edges induced by* $Z_k K_k^{(e)} Z_k P_k^{(e)} e_j$.

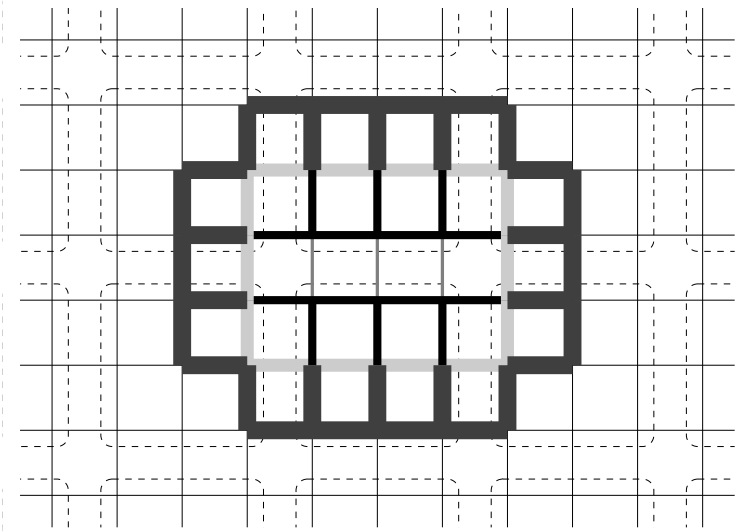


FIG. 4.2. *Nonzero edges due to* $S_k K_k^{(e)} Z_k P_k^{(e)}$ *term.* —— *denotes the three nonzero edges in* $P_k^{(e)} e_j$ *where* e_j *has only one nonzero in the* j^{th} *position.* ▬ *denotes additional nonzero edges in* $Z_k P_k^{(e)} e_j$. ▬ *denotes the extra nonzero edges in* $K_k^{(e)} Z_k P_k^{(e)} e_j$ *and finally* ▬ *denotes the remaining nonzero edges induced by* $S_k K_k^{(e)} Z_k P_k^{(e)} e_j$.

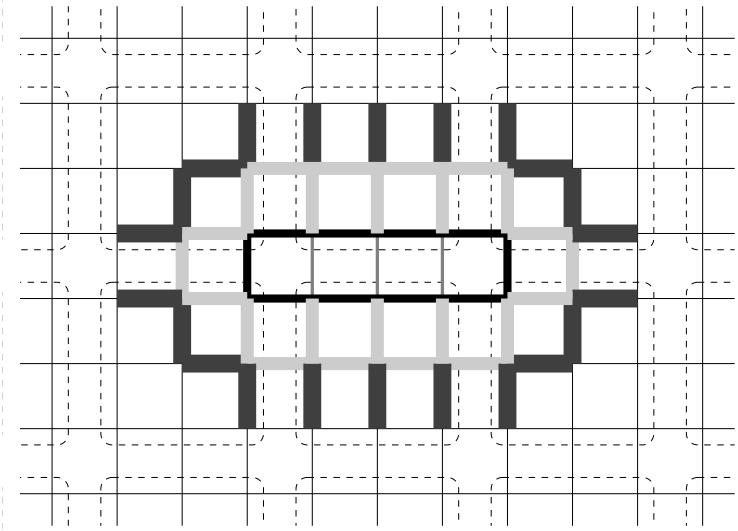


FIG. 4.3. *Nonzero edges due to* $Z_k K_k^{(e)} S_k P_k^{(e)}$ *term.* —— *denotes the three nonzero edges in* $P_k^{(e)} e_j$ *where* e_j *has only one nonzero in the* j^{th} *position.* ▬ *denotes additional nonzero edges in* $S_k P_k^{(e)} e_j$. ▬ *denotes the extra nonzero edges in* $K_k^{(e)} S_k P_k^{(e)} e_j$ *and finally* ▬ *denotes the remaining nonzero edges induced by* $Z_k K_k^{(e)} S_k P_k^{(e)} e_j$.

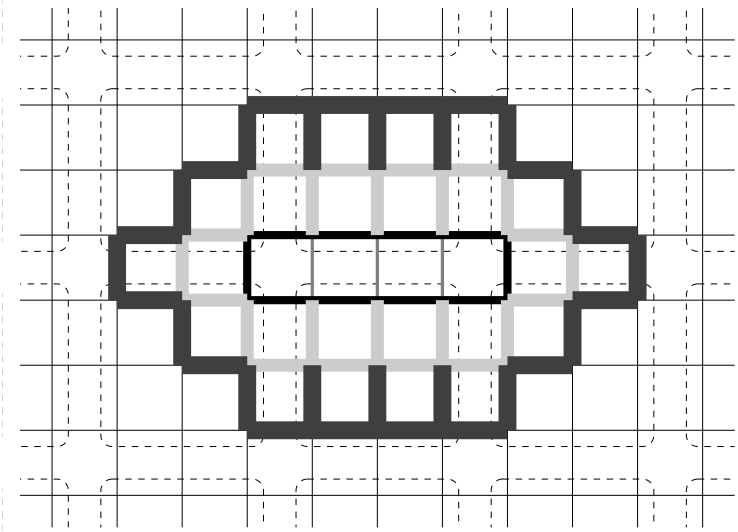


FIG. 4.4. *Nonzero edges due to* $S_k K_k^{(e)} S_k P_k^{(e)}$. *term.* —— *denotes the three nonzero edges in* $P_k^{(e)} e_j$ *where* e_j *has only one nonzero in the* j^{th} *position.* ▬ *denotes additional nonzero edges in* $S_k P_k^{(e)} e_j$. ▬ *denotes the extra nonzero edges in* $K_k^{(e)} S_k P_k^{(e)} e_j$ *and finally* ▬ *denotes the remaining nonzero edges induced by* $S_k K_k^{(e)} S_k P_k^{(e)} e_j$.

grow drastically. Computed complexities will be given with the numerical results to support this claim.

5. Numerical Experiments. To assess the interpolation strategy proposed in this paper, we compare three AMG methods. Each method uses $\mathcal{P}^{(e)}$ with different choices of α and β. See Table 5.1 for the choices of α and β. In each case a multigrid W-cycle is used as a preconditioner for a conjugate gradient iteration. The one exception is the first experiment, where we use both a V- and W-cycle to demonstrate the efficacy of $\mathcal{P}^{(e)}(\alpha,\beta)$. The pre-smoother consists of two steps:

1. A damped Jacobi iteration. For an initial guess, $u^{(0)}$, and right-hand side, f, this gives $u^{(\frac{1}{2})} = u^{(0)} + \mu_1 D^{-1}(f - K_k^{(e)} u^{(0)})$ where D is the diagonal of $K_k^{(e)}$.
2. A damped Jacobi iteration for the projected equations. Defining $\tilde{D}$ as the diagonal of $T_k^T K_k^{(e)} T_k$ gives $u^{(1)} = u^{(\frac{1}{2})} + \mu_2 T_k \tilde{D}^{-1} T_k^T (f - K_k^{(e)} u^{(\frac{1}{2})})$.

These parameters are given by

$$\mu_1 = \frac{60}{31\rho(K_k^{(e)})} \qquad \text{and} \qquad \mu_2 = \frac{60}{31\rho(T_k^T K_k^{(e)} T_k)}.$$

The damping parameters correspond to a first degree Chebyshev polynomial defined over the interval $[\rho(K_k^{(e)})/30, \rho(K_k^{(e)})]$ and $[\rho(T_k^T K_k^{(e)} T_k)/30, \rho(T_k^T K_k^{(e)} T_k)]$, respectively. The intervals effectively define the range of "high frequencies" that must be damped by the smoother. The lower end point is given by dividing the upper end point by the approximate coarsening rate (ratio of unknowns between the fine and coarse meshes) of the multigrid algorithm. Details of this smoother can be found in [1]. It is oriented toward damping high frequency errors and uses only an estimate of the largest eigenvalue of the matrix. The post-smoother performs *step 2* first and then *step 1*, thus guaranteeing the symmetry of the preconditioner. For the uniform-grid examples, a right-hand side is generated by taking a random vector and multiplying by the discretization matrix. A zero initial guess is used for the uniform-grid examples.

$\mathcal{P}^{(e)}(\cdot,\cdot)$	Description of interpolants used in experiments
$\mathcal{P}^{(e)}(0,0)$	Reitzinger and Schöberl interpolation
$\mathcal{P}^{(e)}(\alpha,0)$	Edge smoothed interpolation
$\mathcal{P}^{(e)}(\alpha,\beta)$	Edge and nodal smoothed interpolation

TABLE 5.1
Interpolation used in experiments. Parameters α, β are chosen by (3.9) and (3.10), respectively.

We use the following *T-norm* to evaluate convergence:

$$||r||_T = ||r||_2 + \gamma ||T_1^T r||_2. \tag{5.1}$$

The idea of this norm is to assure that the residual is reduced in both the range space and the null space of the $(curl, curl)$ operator. This is needed to compensate for the poor representation of the error when M_1 is small and the two norm of the residual is used. This can be seen from Equation (2.3). The factor γ in (5.1) is chosen so that the two terms are initially on the same scale. For example, $\gamma = h^2/\sigma$ for Equation (1.1) discretized on a uniform quadrilateral mesh with constant conductivity. In general,

one can choose

$$\gamma = \frac{||K_1^{(e)} v||_2}{||T_1^T K_1^{(e)} v||_2},$$

where $K_1^{(e)} = S_1 + M_1$ and v is a normalized random vector. In this way, we can equilibrate the two terms by the relative sizes of S_1 and M_1. In our experiments the iterations terminate when $||r_{\text{final}}||_T / ||r_0||_T < 10^{-8}$, where r_{final} is the final residual and r_0 is the initial residual.

Before proceeding with the experimental results, there are three practical implementation considerations that should be discussed. First, while in our analysis we assumed that the mass term M_1 is lumped, in practice we do not always lump. As is shown in the experiments, not lumping can result in slightly higher, but still acceptable, multigrid operator complexities. The second is that the mass and stiffness matrices must be projected separately (via the Galerkin product) due to the presence of S_k and $\tilde{M}_k$ in (3.5). This requires additional computation when compared to directly projecting $K_k^{(e)}$ (though this can be significantly reduced if M_k is lumped before projection). There is, however, an important advantage in maintaining the M_k's with respect to numerical cancellation (i.e., rounding errors). Cancellation can be significant when S_k is much larger than M_k and the nodal discretization (needed in the distributed relaxation) is computed via $T_k^T K_k^{(e)} T_k$. This cancellation can be avoided by using $T_k^T M_k T_k$ to compute the nodal discretization. If it is absolutely imperative that S_k and M_k not be stored explicitly (e.g. they are not available on the finest mesh) and it is known that the conductivity is relatively constant, (3.5) can be approximated. In particular, M_k can be ignored (equivalent to approximating it by a multiple of the identity) and $K_k^{(e)}$ can be used instead of S_k without significant convergence degradation.

The third implementation issue concerns the choice of β. We have found in our experimentation that convergence can be enhanced by using a larger β than that given by standard smooth aggregation arguments, (3.10). The reason for this can be seen in (3.6). Specifically, β determines not only the nodal prolongator, (3.2), but also the tentative edge prolongator. While (3.10) is given to optimize the nodal prolongator, it does not necessarily provide for the best tentative edge prolongator. That is, higher values of β can yield a better starting tentative edge prolongators while sacrificing a bit the nodal prolongator. We emphasize that having a good tentative edge prolongator is crucial; for example, just smoothing the original Reitzinger and Schöberl edge prolongator does not yield h-independent multigrid convergence rates. When combined with an appropriate edge smoothing parameter, α, an appropriate choice of β can yield an improved smoothed edge prolongator, $\mathcal{P}^{(e)}$. This difficulty of finding a single β that yields both suitable tentative edge prolongators and smoothed nodal prolongators leads us to suspect that the convergence of the resulting method is not always mesh-independent. We are still exploring alternatives to (3.10) that may produce the best compromise.

5.1. AMG with constant conductivity. We first consider (1.1) discretized with rectangular elements on the unit square with periodic boundary conditions and a constant value of σ. Table 5.2 gives iteration counts and run times for CG preconditioned with a V-cycle. Table 5.3 gives iteration counts and run times for CG preconditioned with a W-cycle. Table 5.4 gives AMG multigrid operator complexity results for the 810×810 mesh.

σ	grid size	$\mathcal{P}^{(e)}(0,0)$	$\mathcal{P}^{(e)}(\alpha,0)$	$\mathcal{P}^{(e)}(\alpha,\beta)$
10^1	30×30	23/0.1	18/0.1	13/0.1
	90×90	40/1.6	26/1.2	16/0.9
	270×270	77/50.5	36/23.4	20/17.1
	810×810	167/1061.9	49/370.6	23/204.3
10^{-1}	30×30	27/0.1	21/0.1	14/0.1
	90×90	47/1.9	33/1.3	18/0.9
	270×270	80/52.7	50/35.1	21/17.2
	810×810	175/1113.5	74/555.7	25/184.4
10^{-3}	30×30	28/0.1	23/0.1	14/0.04
	90×90	48/2.0	35/1.5	18/0.9
	270×270	107/70.1	61/42.6	21/17.7
	810×810	175/1115.0	77/577.7	25/184.7

TABLE 5.2

Iteration counts/run times (in seconds) for the CG/AMG V(1,1) schemes on a 2D model problem with constant conductivity σ.

There are several things to notice about the data in Tables 5.2 and 5.3. The main observation is that AMG using the full $\mathcal{P}^{(e)}(\alpha,\beta)$ operator exhibits near h-independence, as can be seen in Table 5.3. Neither $\mathcal{P}^{(e)}(0,0)$ (the original Reitzinger and Schöberl operator) nor $\mathcal{P}^{(e)}(\alpha,0)$ have h-independent convergence rates, even with W-cycles. Smoothing alone (i.e., $\alpha \neq 0, \beta = 0$) is beneficial but not enough for grid independence. These improved convergence rates translate directly into significantly improved run times (one example is more than five times faster). In addition, for a fixed problem size, the iteration counts of the new prolongator is fairly insensitive to the size of the conductivity for this model problem. This is highly desirable for difficult problems with low conductivity. The operator complexity of the multigrid hierarchy is given in Table 5.4, where the first line corresponds to the multigrid operator complexity for the original Reitzinger and Schöberl algorithm. The complexity for AMG based on $\mathcal{P}^{(e)}(\alpha,\beta)$ requires some explanation. For larger σ, the mass in the third term of (3.1) results in nonzero growth of the coarser level operators. The contribution of the mass term drops with σ. For small σ, these entries can be dropped, and we recover the nonzero pattern of $\mathcal{P}^{(e)}(0,0)$ and $\mathcal{P}^{(e)}(\alpha,0)$. As mentioned in §4, an alternative to using the true mass term is approximating it with a lumped version. Although we did not use a mass approximation in this experiment, doing so has the advantage that there is no growth in the multigrid operator complexities due to the mass term. (See the final slotted-square problem for results using an approximation to the mass matrix.)

Similar results are given in Table 5.5 for a three-dimensional cube. In this case, convergence is independent of the mesh size for the W-cycles, the overall multgrid operator complexity is quite low, and the new prolongator $\mathcal{P}^{(e)}(\alpha,\beta)$ results in significant run time improvements compared to the original prolongator $\mathcal{P}^{(e)}(0,0)$. See Table 5.6 for multigrid operator complexities. We note that in the case of $\mathcal{P}^{(e)}(\alpha,\beta)$ we omitted

σ	grid size	$\mathcal{P}^{(e)}(0,0)$	$\mathcal{P}^{(e)}(\alpha,0)$	$\mathcal{P}^{(e)}(\alpha,\beta)$
10^1	30×30	20/0.1	17/0.1	13 / 0.1
	90×90	32/1.5	21/1.0	15 / 0.8
	270×270	58/39.6	26/19.5	15 / 13.3
	810×810	99/698.3	29/243.0	16 / 134.0
10^{-1}	30×30	23/0.1	20/0.1	13 / 0.1
	90×90	34/1.4	26/1.2	15 / 0.9
	270×270	60/41.1	32/24.8	15 / 14.4
	810×810	105/741.9	35/291.1	16 / 134.2
10^{-3}	30×30	25 / 0.1	21 / 0.1	13 / 0.1
	90×90	34 / 1.4	27 / 1.2	15 / 0.8
	270×270	71 / 49.7	46 / 34.9	15 / 13.7
	810×810	105/739.7	37 / 309.9	16 / 134.0

TABLE 5.3

Iteration counts/run times (in seconds) for the CG/AMG W(1,1) schemes on a 2D model problem with constant conductivity σ.

method	AMG complexity
$\mathcal{P}^{(e)}(0,0)$	1.12
$\mathcal{P}^{(e)}(\alpha,0)$	1.12
$\mathcal{P}^{(e)}(\alpha,\beta)$	1.12 to 1.51

TABLE 5.4

AMG multigrid operator complexities for 2D 810×810 grid, $\sigma \in \{10^1, 10^{-1}, 10^{-3}\}$.

the mass term, M, from the third term of (3.1), i.e., we used only $\beta T D^{-1} T^T$ (due to difficulty in extracting just the mass term within our software). Therefore, we cannot comment on the effect on multigrid operator complexity if the original M were used. If M is lumped, however, clearly there should be no effect on the multigrid operator complexities. We observe h-independent convergence rates for the model problem even without M.

5.2. AMG with conductivity jumps in a 2D domain. For the second experiment, we consider (1.1) discretized with rectangular elements on the unit square, Ω, with periodic boundary conditions. The domain is split into two regions, Ω_1 and Ω_2. Region Ω_1 is given by $[0, 1/3]^2$, and $\Omega_2 = \Omega \setminus \Omega_1$. In Ω_1, the conductivity is either $\sigma_1 = 10^{-3}$ or $\sigma_1 = 10^{-1}$, depending on the experiment. In Ω_2, the conductivity is $\sigma_2 = 1.0$. The domain is depicted in Figure 5.1.

Results are given for a W-cycle in Table 5.7. As with the previous experiments, only the $\mathcal{P}^{(e)}(\alpha,\beta)$ prolongator displays h-independence. There is iteration growth in both $\mathcal{P}^{(e)}(0,0)$ and $\mathcal{P}^{(e)}(\alpha,0)$ as the mesh size increases.

5.3. AMG on a nonorthogonal mesh. As our final computational example, we consider a slotted unit square with a mixture of Neumann and Dirichlet boundary conditions. The exact boundary conditions are given in Figure 5.2. Figure 5.3 illus-

σ	grid size	$\mathcal{P}^{(e)}(0,0)$	$\mathcal{P}^{(e)}(\alpha,0)$	$\mathcal{P}^{(e)}(\alpha,\beta)$
10^1	$15\times15\times15$	19/0.9	19/1.1	19/1.3
	$45\times45\times45$	30/79.7	20/55.5	20/69.9
	$135\times135\times135$	57/5532.7	26/2601.4	19/1968.9†
10^{-1}	$15\times15\times15$	20/1.0	21/1.1	21/1.5
	$45\times45\times45$	34/89.3	22/60.9	21/64.3
	$135\times135\times135$	65/6313.9	29/2913.9	20/2084.2†
10^{-3}	$15\times15\times15$	20/1.0	22/1.1	22/1.5
	$45\times45\times45$	34/85.9	24/64.0	21/61.9
	$135\times135\times135$	78/7526.7	27/2729.5	19/1970.8†

TABLE 5.5

Iteration counts/run times (in seconds) for the CG/AMG W(1,1) schemes on a 3D model problem with constant conductivity σ. †The mass, M, in (3.5) is omitted when defining the edge prolongator.

method	AMG complexity
$\mathcal{P}^{(e)}(0,0)$	1.04
$\mathcal{P}^{(e)}(\alpha,0)$	1.04
$\mathcal{P}^{(e)}(\alpha,\beta)$	1.04†

TABLE 5.6

AMG multigrid operator complexities for 3D $135\times135\times135$ grid, $\sigma\in\{10^1,10^{-1},10^{-3}\}$. (†Note that we omitted the mass M in the third term when defining the edge prolongator.)

trates the computational domain for a mesh with 1,994 edges. Each sphere represents a nodal grid point. The colors of the spheres give an indication of the nodal aggregates used on the finest level. Due to color limitations in the visualization package, it is difficult to entirely distinguish all the aggregates (especially when two aggregates of similar color are adjacent to each other). However, it is clear that the aggregates are certainly not uniform or regular. For these experiments, the following function is used for the conductivity:

$$\sigma = \frac{\eta}{4000}(5-\sin(2\pi xy+3y)e^{(xy-2x)}), \tag{5.2}$$

where η is a scalar and the domain corresponds to $0\le x\le 1$ and $0\le y\le 1$. Figure 5.4 shows a plot of (5.2) for $\eta=10$ on a 33×33 mesh on the unit square. The results given in Table 5.8 illustrate the convergence for different values of η. These results essentially mirror the other experiments given in this paper. In particular, the only convergence rates that are nearly mesh independent correspond to $\mathcal{P}^{(e)}(\alpha,\beta)$.

In these experiments, the mass matrix has many nonzeros (almost as many as the stiffness matrix). This is due to the nonorthogonality of the mesh. To avoid complexity growth, a simple diagonal approximation to the mass matrix is used in (3.5). This diagonal is obtained by simply summing the absolute values of the entries within each row. While this is relatively crude, it seems to be sufficient for these problems. In particular, the diagonal approximation ensures that the multigrid complexity is independent of the mass term, and therefore σ, as discussed in §4. Multigrid complexities

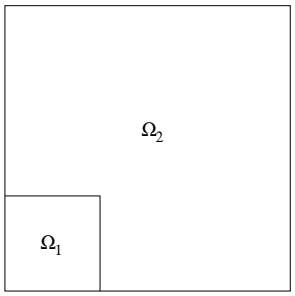


FIG. 5.1. *Domain with two regions of differing conductivity and periodic boundary conditions. In Ω_2 the conductivity is $\sigma_2 = 1.0$. In Ω_1, the conductivity is a constant value, either $\sigma_1 = 10^{-1}$ or 10^{-3}, depending on the experiment.*

σ_1, σ_2	grid size	$\mathcal{P}^{(e)}(0,0)$	$\mathcal{P}^{(e)}(\alpha,0)$	$\mathcal{P}^{(e)}(\alpha,\beta)$
$10^{-1}, 1$	30×30	24/0.1	21/0.1	13/0.1
	90×90	36/1.8	26/1.3	15/1.1
	270×270	62/41.5	31/24.0	15/15.1
	810×810	107/829.8	34/294.9	16/161.1
$10^{-3}, 1$	30×30	24/0.1	21/0.1	14/0.1
	90×90	36/1.8	27/1.4	15/1.1
	270×270	62/41.8	31/23.6	15/16.2
	810×810	107/823.9	35/302.1	16/159.0

TABLE 5.7

Iteration counts/run times (in seconds) for CG/AMG W(1,1) schemes for model problem (1.1) with two regions of different conductivities. The domain is defined in Figure 5.1.

are given in Table 5.9. These complexities are very similar to those observed in the previous experiments.

6. Conclusions. We have proposed a new AMG grid transfer for Maxwell's equations. This operator depends on two parameters that are easy to calculate. Both the Reitzinger and Schöberl intergrid transfers and the smoothed intergrid transfers in [3] are special instances of this new operator. Numerical experiments on model problems show that an AMG method based on this new operator has superior iteration counts and run times to previous operators. In certain cases, the AMG method has h-independent convergence rates. In general, the new AMG/CG method demonstrates improved convergence behavior and has only a slight growth in the number of iterations as the mesh size increases. Future research includes analyzing the behavior of this new method on systems with mass matrices that have coefficient jumps or that have large variation in conductivity.

REFERENCES

[1] M. F. ADAMS, M. BREZINA, J. J. HU, AND R. S. TUMINARO, *Parallel multigrid smoothing: polynomial versus Gauss-Seidel*, Journal of Computational Physics, 188 (2003), pp. 593–610.

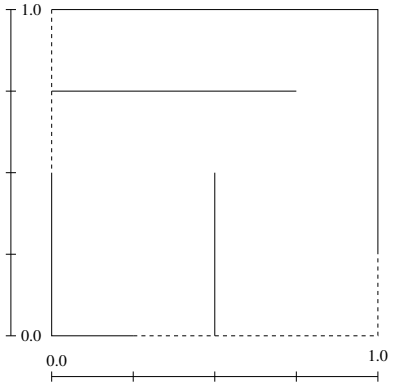


FIG. 5.2. *Boundary conditions for slotted unit square problem. The dashed lines are imposed Neumann conditions, and the solid lines are Dirichlet conditions.*

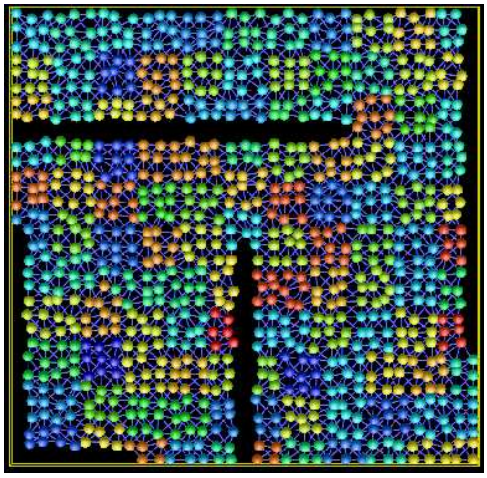


FIG. 5.3. *Visualization of aggregates for* 33×33 *mesh.*

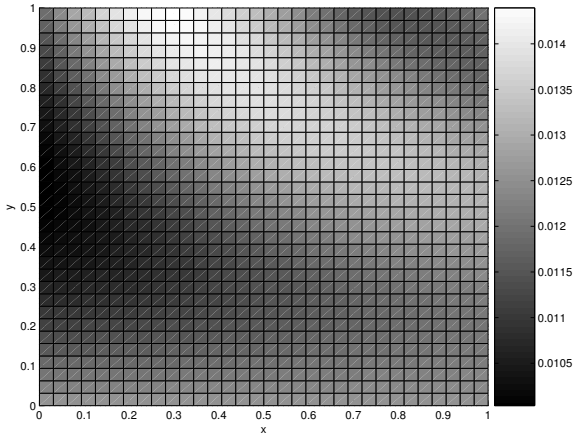


FIG. 5.4. *Variation in* σ*, as given by (5.2), for* $\eta = 10$ *on a* 33×33 *mesh on the unit square.*

η	grid size	$\mathcal{P}^{(e)}(0,0)$	$\mathcal{P}^{(e)}(\alpha,0)$	$\mathcal{P}^{(e)}(\alpha,\beta)$
10^1	$1,994$	24/.09	16/.07	15/.06
	$19,826$	45/1.1	21/.6	16/.5
	$196,643$	86/20.0	27/6.6	17/4.4
	$714,162$	139/125.6	32/30.6	17/17.0
10^{-1}	$1,994$	24/.09	16/.06	15/.06
	$19,826$	45/1.2	21/.6	16/.4
	$196,643$	86/20.0	27/6.6	17/4.5
	$714,162$	139/125.1	32/30.4	17/16.9
10^{-3}	$1,994$	24/.09	16/.06	16/.07
	$19,826$	43/1.1	21/.6	15/.4
	$196,643$	116/26.7	37/8.9	18/4.6
	$714,162$	133/120.2	34/32.0	17/17.0

TABLE 5.8

Iteration counts/run times (in seconds) for CG/AMG W(1,1) schemes for slotted problem. The conductivity is given by (5.2).

grid size	$\mathcal{P}^{(e)}(0,0)$	$\mathcal{P}^{(e)}(\alpha,0)$	$\mathcal{P}^{(e)}(\alpha,\beta)$
$1,994$	1.12	1.17	1.18
$19,826$	1.13	1.16	1.16
$196,643$	1.13	1.13	1.14
$714,162$	1.13	1.13	1.13

TABLE 5.9

Multigrid complexities for slotted problem. Complexities are independent of conductivity.

[2] D. N. ARNOLD, R. S. FALK, AND R. WINTHER, *Multigrid in H(div) and H(curl)*, Numer. Math., 85 (2000), pp. 197–217.

[3] P. B. BOCHEV, C. J. GARASI, J. J. HU, A. C. ROBINSON, AND R. S. TUMINARO, *An improved algebraic multigrid method for solving Maxwell's equations*, Siam Journal on Scientific Computing, 25 (2003), pp. 623–642.

[4] P. B. BOCHEV, J. J. HU, A. C. ROBINSON, AND R. S. TUMINARO, *Towards robust 3D Z-pinch simulations: discretization and fast solvers for magnetic diffusion in heterogeneous conductors*, Electronic Transactions on Numerical Analysis, 15 (2003). http://etna.msc.kent.edu. Special issue for the Tenth Copper Mountain Conference on Multigrid Methods.

[5] A. BOSSAVIT, *A rationale for "edge-elements" in 3-D fields computations*, IEEE Transactions on Magnetics, 24 (1988), pp. 74–79.

[6] J. BRAMBLE, J. PASCIAK, J. WANG, AND J. XU, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp., 57 (1991), pp. 23–45.

[7] A. BRANDT, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, Tech. Report Nr. 85, GMD-Studie, Sankt Augustin, West Germany, 1984.

[8] W. L. BRIGGS, V. E. HENSON, AND S. MCCORMICK, *A multigrid tutorial, Second Edition*, SIAM, Philadelphia, 2000.

[9] W. HACKBUSCH, *Multigrid Methods and Applications*, vol. 4 of Computational Mathematics, Springer–Verlag, Berlin, 1985.

[10] R. Hiptmair, *Multigrid method for Maxwell's equations*, SIAM J. Numer. Anal., 36 (1998), pp. 204–225.

[11] S. Reitzinger and J. Schöberl, *An algebraic multigrid method for finite element discretizations with edge elements*, Numerical Linear Algebra with Applications, 9 (2002), pp. 223–238.

[12] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[13] P. Vaněk, M. Brezina, and J. Mandel, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.

[14] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid based on smoothed aggregation for second and fourth order problems*, Computing, 56 (1996), pp. 179–196.

[15] P. Vassilevski and J. Wang, *Multilevel iterative methods for mixed finite element discretizations of elliptic problems*, Numer. Math., 63 (1992), pp. 503–520.

[16] J. S. V. Welij, *Calculation of eddy currents in terms of H on hexahedra*, IEEE Transactions on Magnetics, MAG-21 (1985), pp. 2239–2241.